IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Matthew Marcus	Art Unit	: 2166
Serial No.	: 10/716,840	Examiner	: Usmaan Saeed
Filed	: November 18, 2003	Conf. No.	: 7089
Title	: OPTIMIZATIONS OF XPATHS		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF

This Reply Brief is filed in response to the Examiner's Answer dated November 20, 2007. Pursuant to 37 C.F.R. § 41.41, Appellant responds to the Examiner's Answer as follows.

1. On page 12, fifth ¶ and in regards to claims 1 and 20 in view of Chau, the Examiner asserted that:

> . . . Examiner respectfully submits that **Chau** teaches **"searching in the hierarchical tree structure only nodes that potentially have child nodes satisfying the Xpath expression"** as the side tables are created by the DAD, and indices are created for columns in the side tables. Therefore, *the search will be fast with indexing*. Note that the invoice_number is the primary key in the application table sales_tab. The advantage of direct query with sub-query is better performance. When *side tables have parent-children relationships*, direct query with sub-query often make more sense (Chau Paragraph 0335 & 0336). [Emphasis in original.]

Whether or not the techniques disclosed in Chau provide fast search with indexing, query with better performance, or query that makes more sense is not at issue. The issue is whether Chau anticipates claim 1. In particular, the Appellant maintains that Chau is not understood to teach at least one element of claim 1: searching in a hierarchical tree structure only nodes that potentially have child nodes satisfying an XPath expression.

On page 13, second ¶, the Examiner cited Chau, paragraphs [0127] and [0214] for teaching populating side tables with values of XML elements or attributes extracted from XML documents and creating indices on the side tables. On page 13, third ¶, the Examiner further asserted that:

> *Examiner interprets that every node in this reference has a potential child nodes. Elements are being extracted from a tree structured XML document and are being stored in the side tables, which have parent-children relationship and these side table are indexed for fast searching using the database B-tree indexing technology.* [Emphasis in original.]

In paragraph [0214], Chau discloses creating an index on the columns of the side tables for fast searching, where the index is created using database B-tree indexing technology. That is, while indexing of the side tables can use database B-tree indexing, searching of the XML document data still involves receiving a query and searching within a main table for elements satisfying the query using indices created on the side tables. *See* Chau, paragraphs [0211], [0214], and [0323]-[0358]. In contrast, the search in claim 1 of the application involves directly searching a hierarchical tree structure of an XML document and searching only nodes that potentially have child nodes satisfying an XPath expression.

2. On page 14, second ¶, and further in regards to claims 1 and 20 in view of Chau, the Examiner relies upon Figure 10 of Chau for teaching that a Document Object Model (DOM) tree comprises relational database nodes. On page 15, second ¶, the Examiner asserted that:

> *Figure 10 shows that the searching is being done by using SQL statements on the desired elements or attributes from the tables, which store nodes of an XML document in a hierarchical tree structure. Examiner interprets that every node in this reference has a potential child nodes.* [Emphasis in original.]

Figure 10 of Chau is a flow diagram illustrating a technique for generating XML documents from data in a relational database using the XPath data model. Part of this method requires creating a DOM, which is an XML document containing a hierarchical tree structure. Chau, paragraphs [0729]-[0730]. The DOM tree comprises relational database nodes, which identify relational tables and columns from which data is to be retrieved. Chau, paragraph [0732].

Chau teaches that "the technique traverses a DOM tree to gather information of each database table to be used in generating one or more XML documents," and the technique will then "generate SQL statements, query relational data [from the relational tables], and write XML document tree contents in a recursive manner" (emphasis added). Chau, paragraph [0731]. The

technique traverses the tree in the DOM to gather information, but it traverses the entire tree. It does not selectively examine only nodes that potentially contain child nodes satisfying an XPath expression. Therefore, it is understood that traversing all nodes in the DOM tree does not teach "searching in the hierarchical tree structure only nodes that potentially have child nodes satisfying the XPath expression," as recited in claim 1.

Furthermore, Chau teaches executing SQL queries of the relational tables to retrieve relational data to be mapped to one or more XML documents. Chau, paragraphs [0731]-[0732]. That is, the queries are executed on the relational tables; a search of the DOM tree nodes is not conducted. In contrast, claim 1 recites searching in a hierarchical tree structure only the nodes that potentially have child nodes satisfying an XPath expression. Hence, Chau's querying of data from the relational tables is not understood to teach searching in a hierarchical tree structure.

For these reasons, and the reasons stated in the Appeal Brief, Appellant submits that the final rejection should be reversed.

Please apply any charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: 1/22/08

Daniel J. Burns
Reg. No. 50,222

Customer No. 21876
Fish & Richardson P.C.
Telephone: (650) 839-5070
Facsimile: (650) 839-5071